

1AP7 Rec'd PCT/PTO 11 JUL 2006

PTO/SB/21 (09-04)


Approved for use through 07/31/2006. OMB 0651-0031

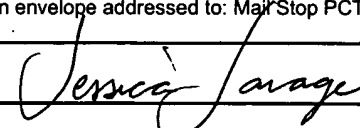
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM <i>(to be used for all correspondence after initial filing)</i>	Application Number	10/564,678	
	Filing Date	13 January 2006	
	First Named Inventor	Sergey Nikolaevich Zheltov	
	Art Unit	Not yet assigned	
	Examiner Name	Not yet assigned	
Total Number of Pages in This Submission	33	Attorney Docket Number	42390P16122

ENCLOSURES (Check all that apply)		
<input type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment/Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input checked="" type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Reply to Missing Parts/Incomplete Application <input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation <input type="checkbox"/> Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____ <input type="checkbox"/> Landscape Table on CD	<input type="checkbox"/> After Allowance Communication to TC <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): Return Receipt Postcard
Remarks: Express Mail No. EV 841 071 825US		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT			
Firm Name	BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP 12400 Wilshire Boulevard, Seventh Floor, Los Angeles, CA 90025-1030		
Signature			
Printed name	James C. Scheller		
Date	7/11/2006	Reg. No.	31,195

CERTIFICATE OF MAILING			
I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.			
Signature			
Typed or printed name	Jessica Savage	Date	7/11/06

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: MS PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

РОСПАТЕНТ
Федеральное государственное учреждение
«Федеральный институт
промышленной собственности
Федеральной службы по интеллектуальной
собственности, патентам и товарным знакам»
(ФГУ ФИПС)
Бережковская наб., 30, корп. 1, Москва, Г-59, ГСП-5, 123995
Телефон 240- 60- 15. Телекс 114818 ПДЧ. Факс 234- 30- 58

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

Наш № 20/12-258

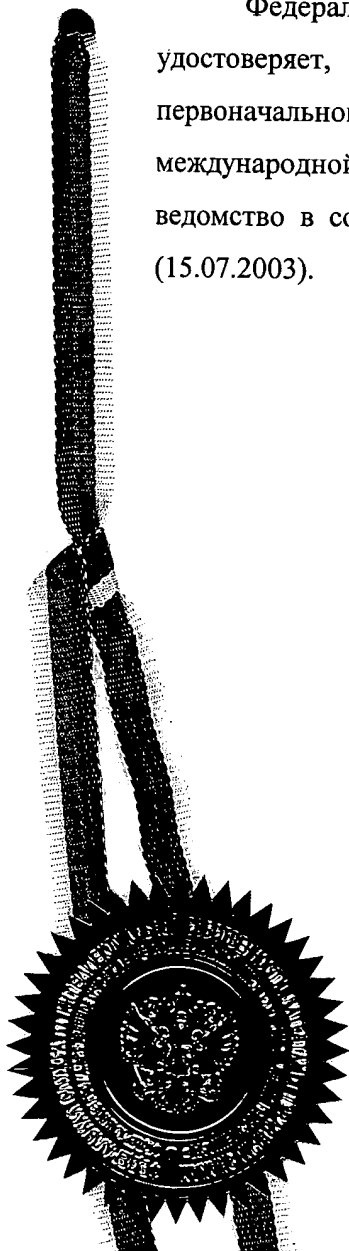
«11» мая 2006 г.

СПРАВКА

Федеральный институт промышленной собственности (далее - Институт) настоящим удостоверяет, что приложенные материалы являются точным воспроизведением первоначального заявления, описания, формулы, реферата и чертежей (если имеются) международной заявки № PCT/RU2003/00307, поданной в Институт как в Получающее ведомство в соответствии с Договором о патентной кооперации 15 июля 2003 года (15.07.2003).

И.О. заведующего отделом 20

Т.Ф.Владимирова



PCT

ЗАЯВЛЕНИЕ

Нижеподписавшийся
просит рассматривать настоящую международную
заявку в соответствии с Договором о патентной
кооперации

Заполняется получающим ведомством	
PCT/RU 0 3 / 0 0 3 0 7	
Номер международной заявки	
15 июля 2003 (15.07.2003)	
Дата международной подачи	
RO/RU	
Наименование получающего ведомства и штамп МЕЖДУНАРОДНАЯ ЗАЯВКА PCT PCT INTERNATIONAL APPLICATION	
№ дела заявителя или агента (по желанию) (максимум 12 знаков)	Z 427

Графа I НАЗВАНИЕ ИЗОБРЕТЕНИЯ "A. method of decoding variable length prefix codes"	
Графа II ЗАЯВИТЕЛЬ	<input type="checkbox"/> Данное лицо является также изобретателем
Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)	
INTEL, ZAKRYTOE AKTSIONERNOE OBSHCHESTVO RU, 125252, Moscow, Sokol-10 Business Center, Chapaevsky per., 14	
Государство (т.е. страна) гражданства: RU	Государство (т.е. страна) местожительства: RU
Данное лицо является заявителем для: <input checked="" type="checkbox"/> всех указанных государств <input checked="" type="checkbox"/> всех указанных государств, кроме США <input type="checkbox"/> только США <input type="checkbox"/> государств, указанных в дополнительной графе	
Графа III ДРУГИЕ ЗАЯВИТЕЛИ И/ИЛИ (ДРУГИЕ) ИЗОБРЕТАТЕЛИ	
Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)	
ZHELTOV Sergei Nikolaevich RU, 603950, Nizhny Novgorod, Turgenev st., 30	
Данное лицо является: <input type="checkbox"/> только заявителем; <input checked="" type="checkbox"/> заявителем и изобретателем <input type="checkbox"/> только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)	
Регистрационный № заявителя в Ведомстве	
Государство (т.е. страна) гражданства: RU	Государство (т.е. страна) местожительства: RU
Данное лицо является заявителем для: <input type="checkbox"/> всех указанных государств <input type="checkbox"/> всех указанных государств, кроме США <input checked="" type="checkbox"/> только США <input type="checkbox"/> государств, указанных в дополнительной графе	
<input checked="" type="checkbox"/> Другие заявители и/или (другие) изобретатели названы на листе продолжения	
Графа IV АГЕНТ ИЛИ ОБЩИЙ ПРЕДСТАВИТЕЛЬ; ИЛИ АДРЕС ДЛЯ ПЕРЕПИСКИ	
Указанное ниже лицо настоящим назначается (назначено) представлять интересы заявителя(ей) в компетентных международных органах в качестве: <input checked="" type="checkbox"/> агента <input type="checkbox"/> общего представителя	
Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны)	
OBSHCHESTVO S OGRANICHENNOI OTVETSTVENNOSTJU "SOJUZPATENT" RU, 103735, Moscow, ul. Iliinka, d.5/2	
Телефон № 925-16-61	
Телефакс № 924-95-40	
Телепринтер № ПОЛУЧЕНО	
Регистрационный № агента в Ведомстве ИИПС ОТЛ №20	
<input type="checkbox"/> Адрес для переписки: Пометить этот бокс, если агент или общий представитель не назначаются (не назначены), а указанный выше адрес используется только как специальный адрес для переписки	

Графа III ДРУГИЕ ЗАЯВИТЕЛИ И/ИЛИ (ДРУГИЕ) ИЗОБРЕТАТЕЛИ*Если ни одна из следующих подграф не используется, этот лист не включается в заявление*

Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таким будет считаться страна указанного в данной графе адреса)

BRATANOV Stanislav Viktorovich
RU, 603950, Nizhny Novgorod,
Turgenev st., 30

Данное лицо является:

☐ только заявителем:☒ заявителем и изобретателем☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

RU

Государство (т.е. страна) местожительства:

RU

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☒

только США

☐государств, указанных в
дополнительной графе

Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таким будет считаться страна указанного в данной графе адреса)

Данное лицо является:

☐ только заявителем:☐ заявителем и изобретателем☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

Государство (т.е. страна) местожительства:

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☐

только США

☐государств, указанных в
дополнительной графе

Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таким будет считаться страна указанного в данной графе адреса)

Данное лицо является:

☐ только заявителем:☐ заявителем и изобретателем☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

Государство (т.е. страна) местожительства:

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☐

только США

☐государств, указанных в
дополнительной графе

Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таким будет считаться страна указанного в данной графе адреса)

Данное лицо является:

☐ только заявителем:☐ заявителем и изобретателем☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

Государство (т.е. страна) местожительства:

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☐

только США

☐государств, указанных в
дополнительной графе☐

Другие заявители и/или (другие) изобретатели названы на другом листе для продолжения

Графа V УКАЗАНИЕ ГОСУДАРСТВ Пометьте нужные боксы ниже, должен быть отмечен как минимум один бокс

Настоящим делаются следующие указания в соответствии с правилом 4.9(a):

Региональный патент

- ☒ **AP** Патент ARIPO: GH Гана, GM Гамбия, KE Кения, LS Лесото, MW Малави, MZ Мозамбик, SD Судан, SL Сьерра-Леоне, SZ Свазиленд, TZ Объединенная Республика Танзания, UG Уганда, ZH Замбия, ZW Зимбабве, а также любое другое государство, являющееся Договаривающимся государством Протокола Хараре и PCT (если испрашивается иной вид охраны или статус, написать на пунктирной линии):
- ☒ **EA** Евразийский патент: AM Армения, AZ Азербайджан, BY Беларусь, KG Кыргызстан, KZ Казахстан, MD Республика Молдова, RU Российская Федерация, TJ Таджикистан, TM Туркменистан, а также любое другое государство, являющееся Договаривающимся государством Евразийской патентной конвенции и PCT
- ☒ **EP** Европейский патент: AT Австрия, BE Бельгия, CH и LI Швейцария и Лихтенштейн, CY Кипр, DE Германия, DK Дания, ES Испания, FI Финляндия, FR Франция, GB Великобритания, GR Греция, IE Ирландия, IT Италия, LU Люксембург, MC Монако, NL Нидерланды, PT Португалия, SE Швеция, TR Турция, а также любое другое государство, являющееся Договаривающимся государством Европейской патентной конвенции и PCT, ~~SI Словения~~
- ☒ **OA** Патент OAPI: BF Буркина Фасо, BJ Бенин, CF Центральная Африканская республика, CG Конго, CI Кот д'Ивуар, CM Камерун, GA Габон, GN Гвинея, GQ Экваториальная Гвинея, GW Гвинея-Бисау, ML Мали, MR Мавритания, NE Нигер, SN Сенегал, TD Чад, TG Того а также любое другое государство, являющееся членом OAPI и Договаривающимся государством PCT (если испрашивается иной вид охраны или статус, написать на пунктирной линии):

Национальный патент (если испрашивается иной вид охраны или статус, написать на пунктирной линии):

- | | | |
|---|---|--|
| <input checked="" type="checkbox"/> AE Объединенные Арабские Эмираты | <input checked="" type="checkbox"/> GM Гамбия | <input checked="" type="checkbox"/> OM Оман |
| <input checked="" type="checkbox"/> AG Антигуа и Барбуда | <input checked="" type="checkbox"/> HR Хорватия | <input checked="" type="checkbox"/> NZ Новая Зеландия |
| <input checked="" type="checkbox"/> AL Албания | <input checked="" type="checkbox"/> HU Венгрия | <input checked="" type="checkbox"/> PH Филиппины |
| <input checked="" type="checkbox"/> AM Армения | <input checked="" type="checkbox"/> ID Индонезия | <input checked="" type="checkbox"/> PL Польша |
| <input checked="" type="checkbox"/> AT Австрия | <input checked="" type="checkbox"/> IL Израиль | <input checked="" type="checkbox"/> PT Португалия |
| <input checked="" type="checkbox"/> AU Австралия | <input checked="" type="checkbox"/> IN Индия | <input checked="" type="checkbox"/> RO Румыния |
| <input checked="" type="checkbox"/> AZ Азербайджан | <input checked="" type="checkbox"/> IS Исландия | <input checked="" type="checkbox"/> RU Российская Федерация |
| <input checked="" type="checkbox"/> BA Босния и Герцеговина | <input checked="" type="checkbox"/> JP Япония | <input checked="" type="checkbox"/> SD Судан |
| <input checked="" type="checkbox"/> BB Барбадос | <input checked="" type="checkbox"/> KE Кения | <input checked="" type="checkbox"/> SE Швеция |
| <input checked="" type="checkbox"/> BG Болгария | <input checked="" type="checkbox"/> KG Кыргызстан | <input checked="" type="checkbox"/> SG Сингапур |
| <input checked="" type="checkbox"/> BR Бразилия | <input checked="" type="checkbox"/> KP Корейская народно-демократическая республика | <input checked="" type="checkbox"/> SI Словения |
| <input checked="" type="checkbox"/> BY Беларусь | <input checked="" type="checkbox"/> KR Республика Корея | <input checked="" type="checkbox"/> SK Словакия |
| <input checked="" type="checkbox"/> BZ Белиз | <input checked="" type="checkbox"/> KZ Казахстан | <input checked="" type="checkbox"/> SL Сьерра-Леоне |
| <input checked="" type="checkbox"/> CA Канада | <input checked="" type="checkbox"/> LC Сент-Люсия | <input checked="" type="checkbox"/> TJ Таджикистан |
| <input checked="" type="checkbox"/> CH and LI Швейцария и Лихтенштейн | <input checked="" type="checkbox"/> LK Шри Ланка | <input checked="" type="checkbox"/> TM Туркменистан |
| <input checked="" type="checkbox"/> CN Китай | <input checked="" type="checkbox"/> LR Либерия | <input checked="" type="checkbox"/> TN Тунис |
| <input checked="" type="checkbox"/> CO Колумбия | <input checked="" type="checkbox"/> LS Лесото | <input checked="" type="checkbox"/> TR Турция |
| <input checked="" type="checkbox"/> CR Коста Рика | <input checked="" type="checkbox"/> LT Литва | <input checked="" type="checkbox"/> TT Тринидад и Тобаго |
| <input checked="" type="checkbox"/> CU Куба | <input checked="" type="checkbox"/> LU Люксембург | <input checked="" type="checkbox"/> TZ Танзания |
| <input checked="" type="checkbox"/> CZ Чешская республика | <input checked="" type="checkbox"/> LV Латвия | <input checked="" type="checkbox"/> UA Украина |
| <input checked="" type="checkbox"/> DE Германия | <input checked="" type="checkbox"/> MA Марокко | <input checked="" type="checkbox"/> UG Уганда |
| <input checked="" type="checkbox"/> DK Дания | <input checked="" type="checkbox"/> MD Республика Молдова | <input checked="" type="checkbox"/> US Соединенные Штаты Америки |
| <input checked="" type="checkbox"/> DM Доминика | <input checked="" type="checkbox"/> MG Малагаскар | <input checked="" type="checkbox"/> UZ Узбекистан |
| <input checked="" type="checkbox"/> DZ Алжир | <input checked="" type="checkbox"/> MK Бывшая Югославская республика Македония | <input checked="" type="checkbox"/> VN Вьетнам |
| <input checked="" type="checkbox"/> EC Эквадор | <input checked="" type="checkbox"/> MN Монголия | <input checked="" type="checkbox"/> YU Югославия |
| <input checked="" type="checkbox"/> EE Эстония | <input checked="" type="checkbox"/> MW Малави | <input checked="" type="checkbox"/> ZA Южная Африка |
| <input checked="" type="checkbox"/> ES Испания | <input checked="" type="checkbox"/> MX Мексика | <input checked="" type="checkbox"/> ZH Замбия |
| <input checked="" type="checkbox"/> FI Финляндия | <input checked="" type="checkbox"/> MZ Мозамбик | <input checked="" type="checkbox"/> ZW Зимбабве |
| <input checked="" type="checkbox"/> GB Великобритания | <input checked="" type="checkbox"/> NO Норвегия | |
| <input checked="" type="checkbox"/> GD Гренада | | |
| <input checked="" type="checkbox"/> GE Грузия | | |
| <input checked="" type="checkbox"/> GH Гана | | |

Боксы, зарезервированные для указания государств, которые стали участниками PCT после выпуска данного листа

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Упоминание о предварительных указаниях: В дополнение к указаниям, сделанным выше, заявитель, в соответствии с правилом 4.9(b), делает также все указания, допустимые в соответствии с PCT, за исключением указания (указаний), приведенного в Дополнительной графе в качестве исключенных из данного упоминания, и заявляет, что эти дополнительные указания подлежат подтверждению, и что любое указание, не подтвержденное до истечения 15 месяцев с даты приоритета, должно считаться изъятым заявителем на момент истечения этого срока. (Подтверждение (включая оплату пошлины) должно быть представлено в получающее ведомство в пределах 15-месячного срока)

Графа VI ЗАЯВЛЕНИЕ НА ПРИОРИТЕТ

Настоящим заявляется приоритет следующей предшествующей заявки(ок) :

Дата подачи предшествующей заявки (день/месяц/год)	Номер предшествующей заявки	Если предшествующая заявка является:		
		национальной заявкой: страна	региональной заявкой: региональное ведомство	международной заявкой: получающее ведомство
(1)				
(2)				
(3)				
(4)				
(5)				

☐ Последующие заявления на приоритет указаны в Дополнительной графе

Получающему ведомству поручается подготовить и направить в Международное бюро заверенную копию предшествующей заявки(заявок)(только в том случае, если предшествующая заявка(заявки) была подана в ведомство, которое для настоящей международной заявки является получающим ведомством), указанную выше как:

☐ все ☐ (1) ☐ (2) ☐ (3) ☐ (4) ☐ (5) ☐ другое, см. Дополнительную графу

*Если предшествующей заявкой является заявка АRIPO, то должна быть указана, по крайней мере, одна страна-участница Парижской конвенции по охране промышленной собственности или одна страна-член Всемирной Торговой Организации, в которую была подана ранняя заявка (правило 4.10(b)(ii)).....

Графа VII МЕЖДУНАРОДНЫЙ ПОИСКОВЫЙ ОРГАН

Выбор международного поискового органа (ISA) (если компетентными в проведении международного поиска являются два или более международных поисковых органа, указать выбранный поисковый орган; можно использовать двубуквенный код):

ISA / RU

Просьба об использовании результатов ранее проведенного поиска; ссылка на такой поиск (если поиск был уже проведен или запрошен у Международного поискового органа ранее):

Дата (день/месяц/год)

Номер

Страна (или региональное ведомство)

Графа VIII ДЕКЛАРАЦИИ

Данное заявление содержит следующие декларации (ниже отметить необходимые боксы и указать в правой колонке количество каждого типа деклараций):

Количество деклараций

- ☐ Графа VIII (i) Декларация об удостоверении личности изобретателя :
- ☐ Графа VIII (ii) Декларация о правомочности заявителя на дату международной подачи подавать заявку и получать патент :
- ☐ Графа VIII (iii) Декларация о правомочности заявителя на дату международной подачи на заявление о приоритете в случае, если он не является заявителем, подавшим предшествующую заявку :
- ☐ Графа VIII (iv) Декларация об авторстве на изобретение для целей указания Соединенных Штатов Америки :
- ☐ Графа VIII (v) Декларация о не наносящих ущерб раскрытиях или изъятиях из-за отсутствия новизны :

Графа IX КОНТРОЛЬНЫЙ ПЕРЕЧЕНЬ; ЯЗЫК ПОДАЧИ

<p>Настоящая международная заявка содержит:</p> <p>(а) следующее количество листов на бумажном носителе:</p> <p>заявление(включая декларацию) : 5</p> <p>описание (исключая перечень последовательностей) : 19</p> <p>формула : 3</p> <p>реферат : 1</p> <p>чертежи : 3</p> <p>Предварительное число листов : 31</p> <p>часть описания с перечнем последовательностей (действительное число листов, представленных на бумажном носителе, независимо от представления в машиночитаемой форме; см. ниже пункт (b)) :</p> <p>Общее число листов : 31</p> <p>(b) перечень последовательностей представлен в машиночитаемой форме</p> <p>(i) <input type="checkbox"/> только (в соответствии с разделом 801(a)(i))</p> <p>(ii) <input type="checkbox"/> как приложение к представленному на бумажном носителе(в соответствии с разделом 801(a)(ii))</p> <p>Тип и количество носителей (дискета, CD-ROM, CD-R или другое), на котором представлен перечень последовательностей (дополнительно к указанному в пункте 9(ii) в правой колонке):</p>	<p>К настоящей международной заявке приложены следующие документы (ниже следует отметить соответствующие боксы и указать с правса количество приложений каждого вида):</p> <p>1. <input type="checkbox"/> лист расчета пошлин</p> <p>2. <input type="checkbox"/> оригинал отдельной доверенности</p> <p>3. <input type="checkbox"/> оригинал генеральной доверенности</p> <p>4. <input type="checkbox"/> копия генеральной доверенности; ссылка на номер, если имеется:.....</p> <p>5. <input type="checkbox"/> разъяснения по поводу отсутствия подписи</p> <p>6. <input type="checkbox"/> приоритетный(ые) документ(ы), указанный в графе VI под №</p> <p>7. <input type="checkbox"/> перевод международной заявки на (язык).....</p> <p>8. <input type="checkbox"/> информация о депонировании микроорганизмов или другого биологического материала</p> <p>9. <input type="checkbox"/> перечень последовательностей в машиночитаемой форме(указать тип и число носителей (дискета, CD-ROM, CD-R или иное))</p> <p>(i) <input type="checkbox"/> копия, представленная для целей международного поиска в соответствии с правилом 13 ter (и не являющаяся частью международной заявки)</p> <p>(ii) <input type="checkbox"/> (только в случае, если слева отмечены бокс(b)(i) или (b)(ii)) дополнительно представленная копия, если допустимо, копия для целей международного поиска в соответствии с правилом 13 ter</p> <p>(iii) <input type="checkbox"/> вместе с соответствующим представлением перечня последовательностей, как его заявление отмечено слева</p> <p>10. <input type="checkbox"/> иное (указать)</p>	<p>Кол-во приложений</p>
<p>Фигура чертежей, предлагаемая для публикации с рефератом: -</p>	<p>Язык подачи международной заявки: GB</p>	

Графа X ПОДПИСЬ ЗАЯВИТЕЛЯ, АГЕНТА ИЛИ ОБЩЕГО ПРЕДСТАВИТЕЛЯ

Рядом с каждой подписью указать фамилию каждого подписавшего и указать, в каком качестве он подписал заявление (если это не очевидно из данных, приведенных в заявлении).

General Director
of OOO "Sojuzpatent"
Felitsyna S.B.

Заполняется получающим ведомством		2. Чертежи:
1. Дата фактического получения международной заявки:	15 июля 2003 (15.07.2003)	<input checked="" type="checkbox"/> получены:
3. Исправленная дата при более позднем, но своевременном получении страниц или чертежей, доукомплектовывающих предполагаемую международную заявку:		<input type="checkbox"/> не получены:
4. Дата своевременного получения требуемых исправлений согласно статье 11(2) PCT:		
5. Международный поисковый орган (если компетентны два и более): ISA/RU	6. <input type="checkbox"/> Направление копии для поиска задержано впредь до уплаты пошлины за поиск	

Заполняется Международным бюро

Дата получения регистрационного экземпляра
Международным бюро:

A Method of Decoding Variable Length Prefix Codes

A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND

1. FIELD

The present invention relates generally to decoding of variable-length prefix codes, e.g., Huffman codes, and, more specifically, to a new, combined decoding scheme of lookup table decoding and prefix oriented decoding.

2. DESCRIPTION

Entropy coding is a widely used data compression technique that many video and audio coding standards are based on. The theoretical basis of entropy coding states that a compression effect can be reached when the most frequently used data are coded with a fewer number of bits than the number of bits denoting the less frequently appearing data. This approach results in coded data streams composed of codes having different lengths.

There are a number of methods to form such variable length codes (VLC). One popular method uses a prefixed coding in which a code consists of a prefix that allows a decoding system to distinguish between different codes, and several significant bits representing a particular value (e.g., Huffman coding).

While most coding standards employ Huffman codes with prefixes composed of a series of '1' or '0' bits in their coding schemes, some standards (e.g., ISO/IEC 14496-2, Moving Pictures Experts Group (MPEG)-4 coding standard, Visual) allow for different coding schemes prefixed with a series of longer bit patterns.

As a general rule, the number of bits that comprise a variable length code depends on the number of bits that comprise the prefix of the code. At the same time, an experimentally defined subset of most frequently appearing codes may have relatively short prefixes (including zero prefix) and, thus, may be decoded in a lookup manner as a single code, which may be a faster way of decoding for a particular system.

Therefore, a need exists for the capability to provide high speed decoding of variable length codes prefixed with regular combinations of bits, in accordance with the actual frequency-to-code length distribution.

5 BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1 is a diagram illustrating an exemplary variable length coding;

10 Figure 2 is a diagram illustrating relations between bits initially read from a bit stream, selected bits, and a table containing a decoded value, a validity indicator and auxiliary information; and

Figure 3 is a flow diagram illustrating the variable length decoding process in accordance with an embodiment of the present invention.

15 DETAILED DESCRIPTION

An embodiment of the present invention is a method of implementing a decoder for variable length codes that have prefixes composed of regular bit patterns. To apply the disclosed method to a particular coding scheme, such a scheme should comprise a subset of most frequently used codes with relatively short prefixes (including zero prefix), such
20 that the prefix scan operation becomes inefficient. According to the disclosed method, the number of bits, not less than the maximal possible length of a VLC, is read from a bit stream. Then a predetermined number of bits is selected and used as an index to a data structure that contains at least a decoded value and validity indicator, along with other pre-decoded data, including but not limited to: prefix type and length, maximal code length for
25 a group of codes, actual code length, and the number of bits to return to the bit stream. The validity indicator is used to determine whether to proceed with the decoding operation, or obtain the valid decoded value from the data structure and return excess bits to the bit stream. If the decoded value is indicated to be invalid, the decoding operation is continued, and a decoding method that estimates the length of the code prefix and the number of
30 significant bits corresponding to the length estimated is applied to the bits initially read from the bit stream. The disclosed method requires less memory than direct lookup decoding methods, and performance of the method exhibits less memory access overhead as compared to prior art methods using multiple lookup tables. Additionally, the present

method appears to be more efficient for decoding of 'short prefix' codes as compared to other prefix oriented methods because it excludes operations of prefix type and length determination for the most frequently used codes.

Reference in the specification to "one embodiment" or "an embodiment" of the present invention means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrase "in one embodiment" appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

Figure 1 is a diagram illustrating an exemplary variable length coding. As depicted by Figure 1, each variable length code has a group of bits used as a prefix 10 and a group of significant bits 12. The prefixes may be composed of a group of bits (bit patterns) that (in a general case) are replicated and concatenated to each other. The bits that follow the code prefix may be called significant bits.

Variable length codes (VLCs) may have identical prefixes. In this case, the codes constitute a prefix code group, but at the same time the number of significant bits that follow the prefix may differ. The maximal number of significant bits that is possible for a code in such a group may be referred to as the maximal bit number. The number of bits that follow the prefix for each VLC may be called the actual bit number.

Figure 2 is a diagram illustrating relations between bits initially read from a bit stream, selected bits, and a table containing a decoded value, a validity indicator and auxiliary information in accordance with an embodiment of the present invention. As depicted in the example of Figure 2, the number of bits 20 not less than any possible VLC length, i.e., the number of bits enough to contain the longest VLC in a particular coding scheme, may be read from a bit stream. Any number of leading bits 22 may be selected from the bits read. A data structure 24 is provided to contain at least decoded data and a validity indicator for each bit combination that may be formed from the selected bits. The data structure 24 may also contain auxiliary information on the type of prefix, code length, and the number of bits to return to the bit stream, in order to facilitate future decoding.

Figure 3 is a flow diagram illustrating a variable length decoding process in accordance with an embodiment of the present invention. At block 100, the number of bits not less than any possible variable length code is read from a bit stream. The number of bits read should be sufficient to contain the longest variable length code but is not limited

to store extra bits as it may facilitate the decoding process (e.g., the bits read fit the machine word size). Then, at block 102, the predetermined number of bits may be selected from the bits previously read. The number of bits to select depends on a particular coding scheme used, and, therefore, is determined by external means. The determination should be performed in a manner that allows the selected bits to span the most frequently used (the most probable) VLCs and at the same time to minimize the size of a code lookup table. At block 104 the code lookup table is indexed with the value formed from the selected bits, and at least a decoded value and a validity indicator, as well as auxiliary information are obtained. In one embodiment, obtaining the auxiliary information may be optional. The validity indicator is then checked at block 106, and if it is indicated to be valid, the decoded value obtained at block 104 is returned as the result of the decoding process at block 108. If necessary, the actual code length or the difference between the actual length and the number of selected bits (retrieved as auxiliary information at block 104) may be checked in order to adjust the bit stream after decoding.

If the decoded data is indicated to be invalid, a prefix oriented decoding method (i.e., a method that estimates the length of the code prefix and the number of significant bits corresponding to the length estimated) is applied to the bits initially read from the bit stream. The auxiliary information obtained at block 104 may describe the type and length of the code prefix, and thus, increase the performance of the method to be further applied.

For an exemplary embodiment of the present invention implemented in the C and Assembler programming languages, refer to Appendix A. This example is non-limiting and one skilled in the art may implement the present invention in other programming languages without departing from the scope of the claimed invention.

The techniques described herein are not limited to any particular hardware or software configuration; they may find applicability in any computing or processing environment. The techniques may be implemented in logic embodied in hardware, software, or firmware components, or a combination of the above. The techniques may be implemented in programs executing on programmable machines such as mobile or stationary computers, personal digital assistants, set top boxes, cellular telephones and pagers, and other electronic devices, that each include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and one or more output devices. Program code is applied to the data entered using the input device to perform the functions described and to

generate output information. The output information may be applied to one or more output devices. One of ordinary skill in the art may appreciate that the invention can be practiced with various computer system configurations, including multiprocessor systems, minicomputers, mainframe computers, and the like. The invention can also be practiced in
5 distributed computing environments where tasks may be performed by remote processing devices that are linked through a communications network.

Each program may be implemented in a high level procedural or object oriented programming language to communicate with a processing system. However, programs may be implemented in assembly or machine language, if desired. In any case, the
10 language may be compiled or interpreted.

Program instructions may be used to cause a general-purpose or special-purpose processing system that is programmed with the instructions to perform the operations described herein. Alternatively, the operations may be performed by specific hardware components that contain hardwired logic for performing the operations, or by any
15 combination of programmed computer components and custom hardware components. The methods described herein may be provided as a computer program product that may include a machine readable medium having stored thereon instructions that may be used to program a processing system or other electronic device to perform the methods. The term "machine readable medium" used herein shall include any medium that is capable of
20 storing or encoding a sequence of instructions for execution by the machine and that cause the machine to perform any one of the methods described herein. The term "machine readable medium" shall accordingly include, but not be limited to, solid-state memories, optical and magnetic disks, and a carrier wave that encodes a data signal. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program,
25 procedure, process, application, module, logic, and so on) as taking an action or causing a result. Such expressions are merely a shorthand way of stating the execution of the software by a processing system cause the processor to perform an action or produce a result.

While this invention has been described with reference to illustrative embodiments,
30 this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.

APPENDIX A

© 2002 Intel Corporation

GetVLC function (Assembler)

5 InitTable function ("C")

Input table ("C") and initial Huffman table (text)

Bit stream structure ("C")

Initial Huffman code table

10 -----

/*

Codes Vector differences

1 0

15 010 1

011 -1

0010 2

0011 -2

00010 3

20 00011 -3

0000110 4

0000111 -4

00001010 5

00001011 -5

25 00001000 6

00001001 -6

00000110 7

00000111 -7

0000010110 8

30 0000010111 -8

0000010100 9

0000010101 -9

0000010010 10

	0000010011	-10
	00000100010	11
	00000100011	-1
	00000100000	12
5	00000100001	-12
	00000011110	13
	00000011111	-13
	00000011100	14
	00000011101	-14
10	00000011010	15
	00000011011	-15
	00000011000	16
	00000011001	-16
	00000010110	17
15	00000010111	-17
	00000010100	18
	00000010101	-18
	00000010010	19
	00000010011	-19
20	00000010000	20
	00000010001	-20
	*/	

25 Packed code/value table containing
information on prefix length and
significant bit number

/// the table elements should be sorted by prefix length

30

```
static const long exTable_Mixed[] =
{
    13, /* max bits | bit-size flag */
```

```

7, /* number of prefix groups */
5, /* lookup table length (in bits) */

1, /* code length */
5 1, /* size of group */
0, /* bit index */
0, /* get bits */
0, /* unget bits */
0x00010000,
10 3, /* 3-bit codes */
2,
1,
1,
0,
15 0x00020001, 0x0003fff,
4, /* 4-bit codes */
2,
2,
1,
20 0,
0x00020002, 0x0003ffe,
5, /* 5-bit codes */
2,
3,
25 1,
0,
0x00020003, 0x0003ffd,
8, /* 8-bit codes */
4,
30 4,
3,
0,
0x00080006, 0x0009ffa, 0x000a0005, 0x000bffb,

```

7, /* 7-bit codes */
2,
4,
3,
5 1,
0x00060004, 0x0007fff, c,
11, /* 11-bit codes */
4,
5,
10 5,
0,
0x0020000c, 0x0021fff4, 0x0022000b, 0x0023fff5,
10, /* 10-bit codes */
6,
15 5,
5,
1,
0x0012000a, 0x0013fff6, 0x00140009, 0x0015fff7, 0x00160008, 0x0017fff8,
8, /* 8-bit codes */
20 2,
5,
5,
3,
0x00060007, 0x0007fff9,
25 11, /* 11-bit codes */
16,
6,
4,
0,
30 0x00100014, 0x0011ffec, 0x00120013, 0x0013ffed, 0x00140012, 0x0015ffee,
0x00160011, 0x0017ffef, 0x00180010, 0x0019fff0, 0x001a000f, 0x001bfff1,
0x001c000e, 0x001dfff2, 0x001e000d, 0x001fff3,


```
-1 /* end of table */
```

```
};
```

```
-----  
5 Bit Stream structures  
-----
```

```
typedef struct _MplDataBuf
```

```
{
```

```
    unsigned char *data;
```

```
10    long      data_len;
```

```
    long      data_offset;
```

```
} MplDataBuf;
```

```
typedef struct _MplBitStream
```

```
15 {
```

```
    long      bit_ptr;      // Buffer bit pointer (31-0)
```

```
    MplDataBuf *data_buf;   // Pointer to data and its size
```

```
20    unsigned long *start_data; // Internal bitsream pointers
```

```
    unsigned long *end_data;
```

```
    unsigned long *current_data;
```

```
25    FILE      *fd;         // Input or output file
```

```
    jmp_buf    exit_point;  // Exit point to handle incorrect vlc codes
```

```
} MplBitStream;
```

```
30 #define DATA_BUFFER_SIZE      1*1024*1024
```

```
unsigned long bit_mask[33] =
```

```
{
```

```

0x00000000,
0x00000001, 0x00000003, 0x00000007, 0x0000000f,
0x0000001f, 0x0000003f, 0x0000007f, 0x000000ff,
0x000001ff, 0x000003ff, 0x000007ff, 0x00000fff,
5 0x00001fff, 0x00003fff, 0x00007fff, 0x0000ffff,
0x0001ffff, 0x0003ffff, 0x0007ffff, 0x000fffff,
0x001fffff, 0x003fffff, 0x007fffff, 0x00ffffff,
0x01ffffff, 0x03ffffff, 0x07ffffff, 0x0fffffff,
0x1fffffff, 0x3fffffff, 0x7fffffff, 0xffffffff
10 };

```

```

-----
Function to form internal VLC table

```

15

```

-----
typedef unsigned long VLCDecodeTable;

```

```

static VLCDecodeTable* CreateVLCDecodeTable_Mixed(const long *src_table,
VLCDecodeTable *table, long *table_size, long cyr_size)

```

20

```

{
    int vm4_vlc_code_mask, vm4_vlc_data_mask, vm4_vlc_shift;
    int offset;
    int i, j;
    int code_length;
    25 int group_size;
    int bit_index;
    int get_bits;
    int unget_bits;
    int group_count;
    30 int outidx;
    int group_offset;
    int lookup_length;
    int prefix_offset;

```

```

switch(*src_table++ & VM4_VLC_LEN_FLAG)
{
case VM4_VLC_20:
    vm4_vlc_code_mask = 0xffff000;
5    vm4_vlc_data_mask = 0x00000fff;
    vm4_vlc_shift = 12;
    break;
case VM4_VLC_24:
    vm4_vlc_code_mask = 0xffff00;
10    vm4_vlc_data_mask = 0x000000ff;
    vm4_vlc_shift = 8;
    break;
default:
    vm4_vlc_code_mask = 0xffff0000;
15    vm4_vlc_data_mask = 0x00000fff;
    vm4_vlc_shift = 16;
    break;
}

20    offset = *src_table++ * 2;
    lookup_length = *src_table++;
    prefix_offset = (1 << lookup_length) * 2 + 2;
    offset += prefix_offset;

25    memset(table, 0, offset * sizeof(VLCDecodeTable));
    ///memset(table, -1, prefix_offset * sizeof(VLCDecodeTable));

    table[0] = 32 - lookup_length; /// the bit count to shift right
    table[1] = prefix_offset;

30    while(*src_table != -1)
    {
        code_length = *src_table++;

```

```

group_size = *src_table++;
bit_index  = *src_table++ * 2 + prefix_offset;
get_bits   = *src_table++;
unget_bits = *src_table++;

5
if(!table[bit_index])
{
    table[bit_index]  = get_bits;
    table[bit_index + 1] = group_offset = offset;
10
}
for(i = 0, group_count = 0; i < group_size; i++)
{

    if(code_length < lookup_length)
15
    {
        for(j = 0; j < (1 << (lookup_length - code_length)); j++)
        {
            outidx = ((((((unsigned long int)(*src_table & vm4_vlc_code_mask))
20
                >> vm4_vlc_shift) & bit_mask[code_length])
                << (lookup_length - code_length)) + j) * 2;

            table[outidx + 2]  = /*lookup_length - */code_length;
            table[outidx + 2 + 1] = ((*src_table & vm4_vlc_data_mask) << (32 -
                vm4_vlc_shift)) >> (32 - vm4_vlc_shift);
25
        }
    }
    else if(code_length == lookup_length)
    {
        outidx = ((((((unsigned long int)(*src_table & vm4_vlc_code_mask))
30
            >> vm4_vlc_shift) & bit_mask[code_length]) * 2;

        table[outidx + 2]  = code_length; //0;
        table[outidx + 2 + 1] = ((*src_table & vm4_vlc_data_mask) << (32 -

```

14

```

        vm4_vlc_shift)) >> (32 - vm4_vlc_shift);
    }

    if(!unget_bits)
5    {
        outidx = (((((unsigned long int)(*src_table & vm4_vlc_code_mask))
            >> vm4_vlc_shift) & bit_mask[get_bits]) * 2;

        table[group_offset + outidx] = ((*src_table & vm4_vlc_data_mask) <<
10            (32 - vm4_vlc_shift)) >> (32 -
            vm4_vlc_shift);
        table[group_offset + outidx + 1] = 0;
        group_count++;
        src_table++;
15    }
    else
    {
        for(j = 0; j < (1 << unget_bits); j++)
        {
20            outidx = ((((((unsigned long int)(*src_table & vm4_vlc_code_mask))
                >> vm4_vlc_shift) & bit_mask[get_bits - unget_bits])
                << unget_bits) + j) * 2;

            table[group_offset + outidx] = ((*src_table & vm4_vlc_data_mask)
25            << (32 - vm4_vlc_shift)) >> (32 -
                vm4_vlc_shift);
            table[group_offset + outidx + 1] = unget_bits;
            group_count++;
        }
30        src_table++;
    }
}

offset += group_count * 2;

```

15

}

*table_size = offset;

```

5      return (VLCDDecodeTable*)table;
      }

```

```

10      -----
      Function to decode VLC (Assembler)
      -----

```

.686

.xmm

xmmword textequ <qword>

```

15      mmword      textequ <qword>

```

.model FLAT

MplDataBuf struc 4t

data dd ?

```

20      data_len dd ?

```

data_offset dd ?

MplDataBuf ends

MplBitStream struc 4t

```

25      bit_ptr dd ? ;;; Buffer bit pointer (31-0)

```

```

      data_buf dd ? ;;; Pointer to data and its size

```

```

      start_data dd ? ;;; Internal bitsream pointers

```

```

30      end_data dd ?

```

current_data dd ?

```

      fd dd ? ;;; Input or output file

```

```

exit_point    dd    ?        ;;; Exit point to handle incorrect vlc codes
MplBitStream ends

```

```

5      _TEXT      segment

```

```

        extrn    _longjmp:near

```

```

        ;;; unsigned long asmbsGetVLC_LookupBitSearch
10      ;;;
        (MplBitStream *bsm, const VLCDecodeTable *vlcTable)

```

```

_asmbsGetVLC_LookupBitSearch  proc    near

```

```

    sizeof_locals  equ    14h

```

```

    ws      equ     esp + 04h

```

```

15

```

```

    bsm      equ     dword ptr [eax + 04h]

```

```

    table    equ     dword ptr [eax + 08h]

```

```

        mov     eax,esp

```

```

20      sub     esp,sizeof_locals

```

```

        and     esp,0ffffff0h

```

```

        push    eax

```

```

        mov     [ws],esi

```

```

        mov     [ws + 04h],edi

```

```

25      mov     [ws + 08h],ecx

```

```

        mov     [ws + 0ch],ebx

```

```

        mov     [ws + 10h],ebp

```

```

        mov     esi,bsm

```

```

        mov     edi,table

```

```

30

```

```

        mov     ecx,1fh

```

```

        sub     ecx,MplBitStream.bit_ptr[esi]

```

```

        mov     ebx,MplBitStream.current_data[esi]

```

17

```

    mov     eax,[ebx]
    mov     edx,[ebx + 4]
    shld    eax,edx,cl          ;;; eax = data

5      test    eax,eax
      jz     error_code        ;;; this branch is supposed not to be taken

    ;;; look up several bits first
    mov     ecx,[edi]          ;;; ecx == 32 - lookup_bits
10     mov     edx,eax
      shr     edx,cl
    mov     ebp,[edi + edx * 8 + 8] ;;; ebp == (un)get bits
    or      ebp,ebp
    jz      scan              ;;; not taken
15     mov     eax,[edi + edx * 8 + 0ch] ;;; eax == decoded data
    mov     ebx,MplBitStream.bit_ptr[esi]
    sub     ebx,ebp
    js      negative_ptr      ;;; not taken

20     ;;; exit
    mov     MplBitStream.bit_ptr[esi],ebx
    mov     esi,[ws]
    mov     edi,[ws + 04h]
    mov     ecx,[ws + 08h]
25     mov     ebx,[ws + 0ch]
    mov     ebp,[ws + 10h]
    mov     esp,[esp]
    ret

30     scan:
      bsr     ecx,eax          ;;; ecx = index

    mov     ebx,[edi + 4]      ;;; ebx == prefix_offset

```


18

```

    add    ebx,62
    mov    ebp,31
    sub    ebx,ecx
    sub    ebx,ecx                ;;; ebx = offset (of bit index group)
5   sub    ebp,ecx                ;;; ebp = (31 - index)
    mov    edx,[edi + ebx * 4]    ;;; edx = get_bits
    mov    ebx,[edi + ebx * 4 + 4];;; ebx = offset (of code value and unget bits)

    sub    ecx,edx
10   shr    eax,cl
    and    eax,bit_mask[edx * 4];;; eax = data

    lea    ebx,[ebx * 4]
    lea    ebx,[ebx + eax * 8]
15   mov    ecx,[edi + ebx + 4]    ;;; ecx = unget_bits
    mov    eax,[edi + ebx]        ;;; eax = data

    mov    ebx,MplBitStream.bit_ptr[esi]
    lea    edx,[edx + ebp + 1]
20   add    ebx,ecx
    sub    ebx,edx

    js     negative_ptr          ;;; not taken

25   almost_exit:
        mov    MplBitStream.bit_ptr[esi],ebx

    exit:

    mov    esi,[ws]
30   mov    edi,[ws + 04h]
    mov    ecx,[ws + 08h]
    mov    ebx,[ws + 0ch]
    mov    ebp,[ws + 10h]

```

19

```

mov    esp,[esp]
ret

```

```

negative_ptr:

```

```

5      add    ebx,20h
      add    MplBitStream.current_data[esi],04h
      jmp    almost_exit      ;;; taken

```

```

error_code:

```

```

10     push   -1
      lea     edx,MplBitStream.exit_point[esi]
      push   edx
      call    _longjmp
      ;;; no return here
15     int     00h

```

```

_asmbsGetVLC_LookupBitSearch    endp

```

```

_TEXT    ends

```

```

_DATA    segment

```

```

bit_mask    dd    00000000h
            dd    00000001h, 00000003h, 00000007h, 0000000fh
25          dd    0000001fh, 0000003fh, 0000007fh, 000000ffh
            dd    000001ffh, 000003ffh, 000007ffh, 00000fffh
            dd    00001fffh, 00003fffh, 00007fffh, 0000ffffh
            dd    0001ffffh, 0003ffffh, 0007ffffh, 000ffffh
            dd    001ffffh, 003ffffh, 007ffffh, 00ffffffh
30          dd    01ffffh, 03ffffh, 07ffffh, 0ffffffh
            dd    1ffffh, 3ffffh, 7ffffh, 0ffffffh

```

```

_DATA    ends

```

```

end

```

CLAIMS

What is claimed is:

- 5 1. In a system for decoding variable length prefix codes in a bit stream, a method comprising:
reading, from the bit stream, a number of bits sufficient to store a longest variable
length code of the system;
selecting a predetermined number of bits from the bits read; and
10 obtaining, from a data structure, in accordance with an actual value of the bits
selected, at least a decoded value and a validity indicator associated with a variable length
code.
- 15 2. The method of claim 1, further comprising applying a prefix oriented decoding
method to the bits initially read from the bit stream when the decoded value is indicated to
be invalid.
3. The method of claim 1, wherein reading the number of bits comprises making
the specified number of bits accessible for future operations.
- 20 4. The method of claim 1, wherein selecting the number of bits comprises making
the specified number of bits accessible for future operations, faster than reading the same
number of bits.
5. The method of claim 1, wherein the predetermined number of bits comprises the
maximal number of bits to be used as an index to the data structure.
6. The method of claim 1, wherein the validity indicator indicates whether the
decoded value is valid.
- 25 7. The method of claim 1, wherein the data structure used to obtain at least the
decoded value and validity indicator associated with a variable length code comprises a
memory area containing at least the decoded value and validity indicator for each bit
combination that can be formed from the predetermined number of bits.
- 30 8. The method of claim 2, wherein the prefix oriented decoding method further
comprises a method of variable length decoding that employs variable length code prefix
properties during decoding.

9. An article comprising: a machine accessible medium having a plurality of machine readable instructions, wherein when the instructions are executed by a processor, the instructions provide for decoding of variable length prefix codes in a bit stream by

5 reading, from the bit stream, a number of bits sufficient to store a longest variable length code of the system;

selecting a predetermined number of bits from the bits read; and

obtaining, from a data structure, in accordance with an actual value of the bits selected, at least a decoded value and validity indicator associated with a variable length code.

10 10. The article of claim 9, further comprising instructions for applying a prefix oriented decoding method to the bits initially read from the bit stream when the decoded value is indicated to be invalid.

11. The article of claim 9, wherein instructions for reading the number of bits comprise instructions for making the specified number of bits accessible for future
15 operations.

12. The article of claim 9, wherein instructions for selecting the number of bits comprise instructions for making the specified number of bits accessible for future operations, faster than reading the same number of bits.

13. The article of claim 9, wherein the predetermined number of bits comprises the
20 maximal number of bits to be used as an index to the data structure.

14. The article of claim 9, wherein the validity indicator indicates whether the decoded value is valid.

15. The article of claim 9, wherein the data structure used to obtain at least the decoded value and validity indicator associated with a variable length code comprises a
25 memory area containing at least the decoded value and validity indicator for each bit combination that can be formed from the predetermined number of bits.

16. The article of claim 10, wherein prefix oriented decoding method further comprises a method of variable length decoding that employs variable length code prefix properties during decoding.

30 17. A system for decoding variable length prefix codes in a bit stream, comprising:
logic to read from the bit stream a number of bits sufficient to store a longest variable length code of the system;

logic to select a predetermined number of bits from the bits read; and

logic to obtain from a data structure, in accordance with the actual value of the bits selected, at least a decoded value and a validity indicator associated with a variable length code.

5 18. The system of claim 17, further comprising logic to apply a prefix oriented decoding method to the bits initially read from the bit stream when the decoded value is indicated to be invalid.

19. The system of claim 17, wherein logic to read the number of bits comprises logic to make the specified number of bits accessible for future operations.

10 20. The system of claim 17, wherein logic to select the number of bits comprises logic to make the specified number of bits accessible for future operations, faster than logic to read the same number of bits.

21. The system of claim 17, wherein the predetermined number of bits comprises the maximal number of bits to be used as an index to the data structure.

15 22. The system of claim 17, wherein the validity indicator indicates whether the decoded value is valid.

23. The system of claim 17, wherein the data structure used to obtain at least the decoded value and validity indicator associated with a variable length code comprises a memory area containing at least the decoded value and validity indicator for each bit combination that can be formed from the predetermined number of bits.

20 24. The system of claim 18, wherein prefix oriented decoding method further comprises a method of variable length decoding that employs variable length code prefix properties during decoding.

25

30

ABSTRACT OF THE DISCLOSURE

The method disclosed may be used together with any prefix oriented decoding method to enable faster decoding of variable length codes when a subset of most frequently
5 used codes with relatively short prefixes may be determined. An embodiment of the present invention reads a number of bits, not less than the maximal possible length of a code, from a bit stream. Then a predetermined number of bits is selected and used as an index to a data structure that contains at least a decoded value and a validity indicator, along with other pre-decoded data, namely: prefix type and length, maximal code length
10 for a group of codes, actual code length, the number of bits to return to the bit stream, etc. The validity indicator is used to determine whether to proceed with the decoding operation, or obtain the valid decoded value from the data structure and return excess bits to the bit stream. If the decoded value is indicated to be invalid, the decoding operation is continued, and a decoding method that estimates the length of the code prefix and the number of
15 significant bits corresponding to the length estimated is applied to the bits initially read from the bit stream.

1/3

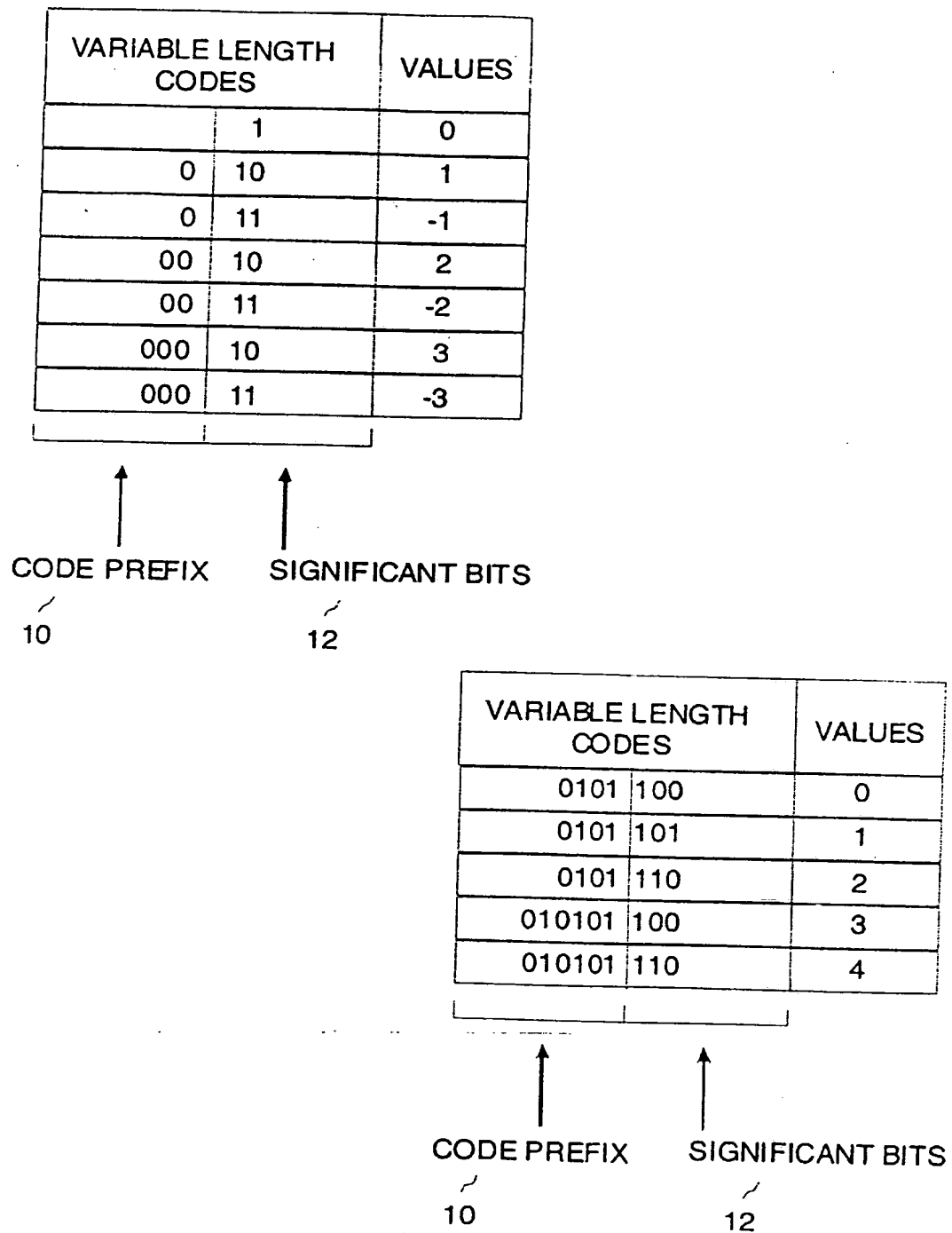


FIG. 1

2/3

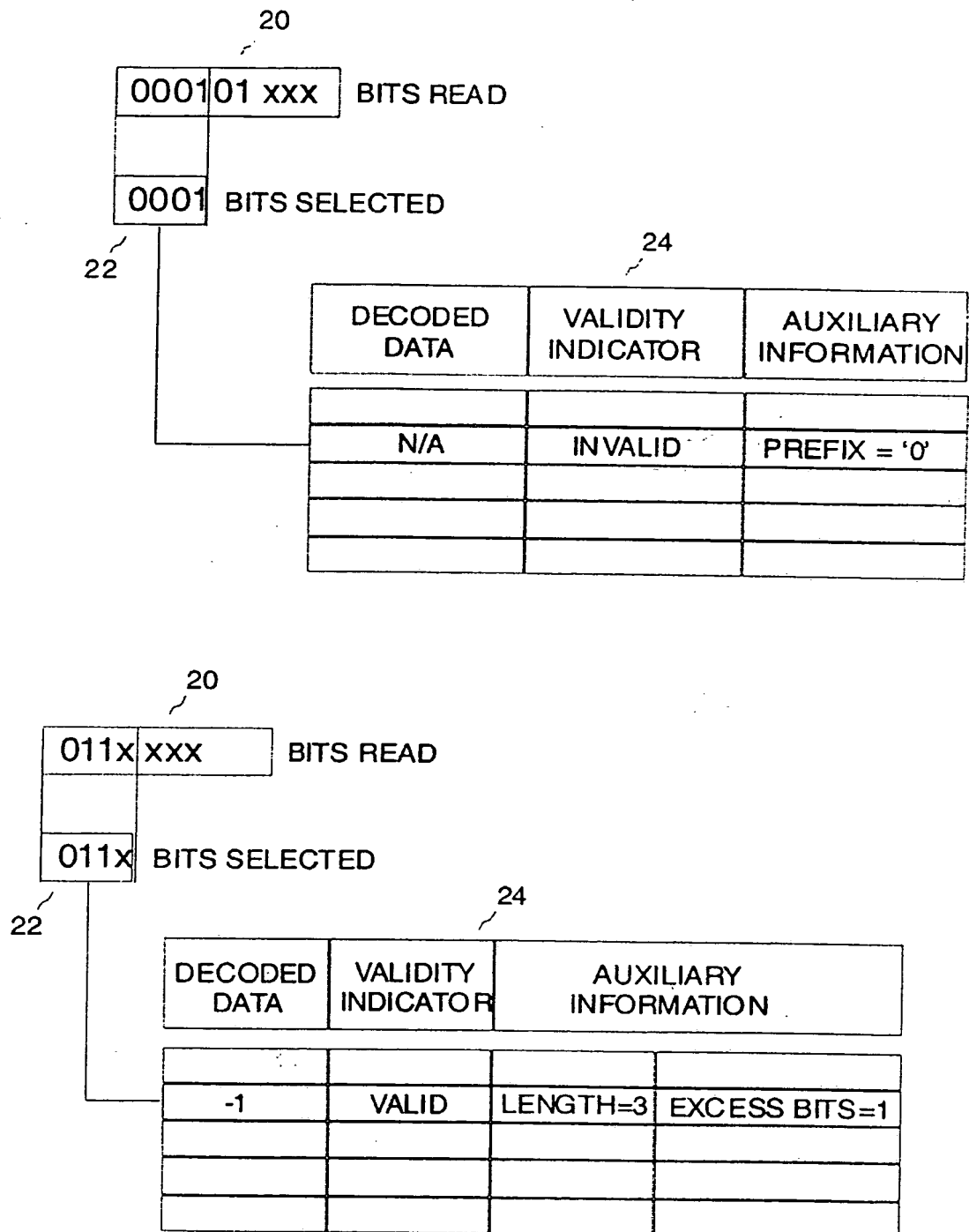


FIG. 2

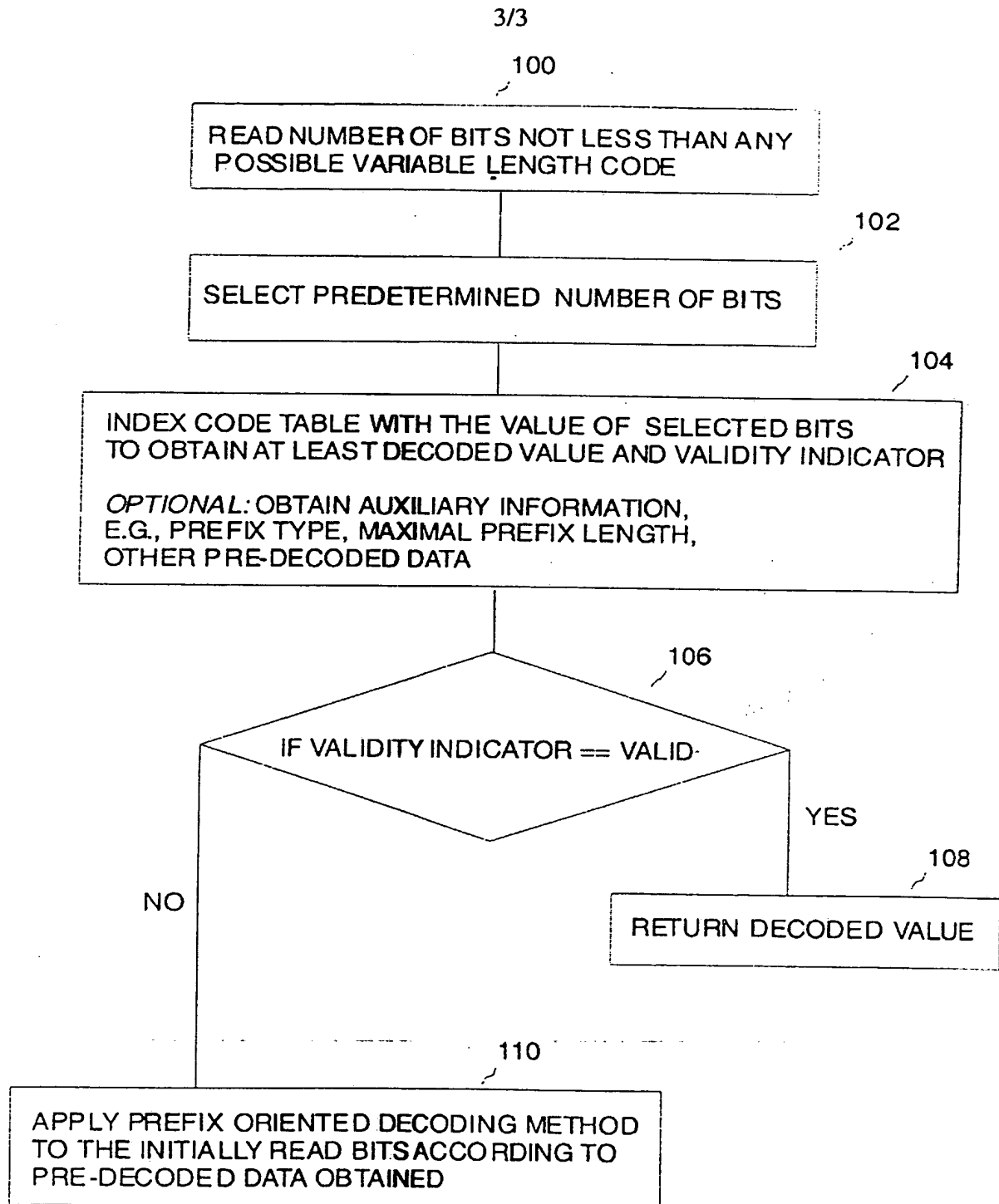


FIG. 3